IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: James McKeeth	§	Group Art Unit: 2191
Application No.: 09/449,782	§	Confirmation No: 6698
Filed: November 26, 1999	§	Examiner: Brophy, Matthew J.
For: COMMAND LINE OUTPUT REDIRECTION	§	Atty. Docket: MICS:0194/FLE/MAN/TOM (MUEI-0531.00/US)

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF TRANSMISSION OR MAILING
37 C.F.R. 1.8

I hereby certify that this correspondence is being transmitted by facsimile to the United States Patent and Trademark Office in accordance with 37 C.F.R. § 1.6(d), or is being transmitted via the Office electronic filing system in accordance with 37 C.F.R. § 1.6(a)(4), or is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date below:

October 26, 2010	**/Robert A. Manware/**
Date	Robert A. Manware

APPEAL BRIEF PURSUANT TO 37 C.F.R. §§ 41.31 AND 41.37

This Appeal Brief is being filed in furtherance of the Notice of Appeal electronically filed on August 26, 2010. The Commissioner is authorized to charge the requisite fee of $540.00, and any additional fees which may be required, to the credit card charge authorization submitted electronically with the present filing. However, if for any reason this charge fails, the Commissioner is authorized to charge Deposit Account No. 06-1315; Order No. MICS:0194/FLE/MAN (MUEI-0531.00/US).

1. REAL PARTY IN INTEREST

The real party in interest is Micron Electronics, Inc., the Assignee of the above-referenced application by virtue of the Assignment recorded at reel 010418, frame 0489,

and dated November 26, 1999. Accordingly, Micron Electronics, Inc. will be directly affected by the Board's decision in the pending appeal.

2. **RELATED APPEALS AND INTERFERENCES**

Appellant is unaware of any other appeals or interferences related to this Appeal. The undersigned is Appellant's legal representative in this Appeal.

3. **STATUS OF CLAIMS**

Claims 1-5, 7, 10-15, 18-21 and 23-28 are currently pending, are currently under final rejection and, thus, are the subject of this Appeal. Claims 6, 8, 9, 16, 17 and 22 were previously canceled.

4. **STATUS OF AMENDMENTS**

As the instant claims have not been amended since the mailing of the Final Office Action, there are no outstanding amendments to be considered by the Board.

5. **SUMMARY OF CLAIMED SUBJECT MATTER**

The present invention relates generally to the field of computer systems and, more particularly, to redirecting command line utility output to a non-application maintained storage location. The Application contains three independent claims, namely, claims 1, 15, and 21, all of which are the subject of this Appeal. The subject matter of these claims is summarized below.

With regard to independent claim 1, discussions of the recited features of claim 1 can be found at least in the below cited locations of the specification and drawings. By way of example, a method (e.g., redirection routine 200) includes invoking, by an application, a call of a command line utility, the application providing an identifier in the call of the command line utility, wherein the command line utility is a utility executable from a command line prompt (e.g., block 202), receiving output from the command line utility (e.g., block 204), storing the command line utility output in a system registry

database at a location identified by the identifier (e.g. block 206), and retrieving, by the application, the command line utility output from the system registry at the location identified by the identifier. *See* e.g., Application, page 3, lines 9-30; page 4, lines 1-30; page 5, lines 1-30, page 6, lines 1-2; Fig. 2.

With regard to independent claim 15, discussions of the recited features of claim 15 can be found at least in the below cited locations of the specification and drawings. By way of example, a program storage device (e.g., storage device 404), readable by a computer (e.g., computer system 400), includes instructions (e.g., routine 402) stored on the program storage device for causing the computer to cause an application to invoke a call of a command line utility, the application providing an identifier in the call of the command utility, wherein the command line utility is a utility executable from a command line prompt. *See* e.g., Application, page 3, lines 9-30; page 4, lines 1-30; page 5, lines 1-30, page 6, lines 1-2; page 7, lines 2-8; Figs. 2 and 4. The instructions further cause the computer to receive output from the command line utility, store the command line utility output in a system registry database at a location identified by the identifier, and cause the application to retrieve the command line utility output from the system registry database or shared system memory at the location identified by the identifier. *See* e.g., *id*., page 3, lines 9-30; page 4, lines 1-30; page 5, lines 1-30, page 6, lines 1-2; Fig. 2.

With regard to independent claim 21, discussions of the recited features of claim 21 can be found at least in the below cited locations of the specification and drawings. By way of example, a computer system (e.g., computer 400), that includes a processor (e.g., processor 406), a command line utility, wherein the command line utility is a utility executable from a command line prompt, an application executable on the processor, the application to call the command line utility, the application to provide an identifier in the call, and a system registry database having a location identified by the identifier, the location identified by the identifier to store an output of the command line utility, the application to retrieve the command line utility output from the location identified by the

identifier. *See* e.g., Application, page 3, lines 9-30; page 4, lines 1-30; page 5, lines 1-30, page 6, lines 1-2; Fig. 2

6. **GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL**

First Ground of Rejection for Review on Appeal:

Appellant respectfully urges the Board to review and reverse the Examiner's first ground of rejection in which the Examiner rejected claims 1, 5, 10, 11, 15, 18, 21, and 23-28 under 35 U.S.C. § 103(a) as being unpatentable over Appellant's Admitted Prior Art (hereinafter "AAPA") in view of "The Windows NT Command Shell" by Tim Hill (hereinafter "Hill") in view of Buxton (U.S. Patent No. 6,182,279, hereinafter "Buxton") and further in view of Hlava (US Patent 6,681,265, hereinafter "Hlava").

Second Ground of Rejection for Review on Appeal:

Appellant respectfully urges the Board to review and reverse the Examiner's second ground of rejection in which the Examiner rejected claims 2-4, 7, 12-14, 19 and 20 under 35 U.S.C. § 103(a) as being unpatentable over AAPA in view of Hill in view of Hlava as applied above and further in view of Buxton.

7. **ARGUMENT**

As discussed in detail below, the Examiner has improperly rejected the pending claims. Further, the Examiner has misapplied long-standing and binding legal precedents and principles in rejecting the claims under 35 U.S.C. § 103. Accordingly, Appellant respectfully requests full and favorable consideration by the Board, as Appellant strongly believes that claims 1-5, 7, 10-15, 18-21, and 23-28 are currently in condition for allowance.

A. **Ground of Rejection No. 1:**

The Examiner rejected claims 1, 5, 10, 11, 15, 18, 21, and 23-28 under 35 U.S.C. § 103(a) as being unpatentable over AAPA in view of Hill in view of Buxton and further in view of Hlava.

1. **Judicial precedent has clearly established a legal standard for a *prima facie* obviousness rejection.**

The burden of establishing a *prima facie* case of obviousness falls on the Examiner. *Ex parte Wolters and Kuypers*, 214 U.S.P.Q. 735 (B.P.A.I. 1979). To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 180 U.S.P.Q. 580 (C.C.P.A. 1974). However, a claimed invention composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art. *KSR International Co. v. Teleflex Inc.*, 127 S.Ct. 1727, 1741 (2007). The *KSR* court stated that "it can be important to identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the elements in the way the claimed new invention does … because inventions in most, if not all, instances rely upon building blocks long since uncovered, and claimed discoveries almost of necessity will be combinations of what, in some sense, is already known." *Id.* Specifically, there must be some articulated reasoning with a rational underpinning to support a conclusion of obviousness; a conclusory statement will not suffice. *In re Kahn,* 441 F.3d 977, 988 (Fed. Cir. 2006). Indeed, the factual inquiry determining whether to combine references must be thorough and searching, and it must be based on objective evidence of record. *In re Lee,* 61 U.S.P.Q.2d 1430, 1436 (Fed. Cir. 2002).

2. **The Examiner's rejection of independent claims 1, 15, and 21 is improper because the rejection fails to establish a *prima facie* case of obviousness with regard to claims 1, 15, and 21.**

Independent claim 1 recites "storing the command line utility output in a system registry database at a location identified by the identifier." Similarly, independent claim

15 recites "instructions for causing the computer to … store the command line utility output in a system registry database at a location identified by the identifier." Finally, independent claim 21 recites "a system registry database having a location identified by the identifier, the location identified by the identifier to store an output of the command line utility."

In the rejection, the Examiner cited AAPA as disclosing storing "command line utility output … at a location" and retrieving "the command line utility output … at the location identified by the identifier." However, the Examiner admitted that AAPA does not disclose a "system registry database." Instead, the Examiner cited Halva as disclosing storing and retrieving to and from a "system registry database." However, Appellant asserts that the Examiner has not met the burden of showing a *prima facie* case of obviousness of claims 1, 15, and 21. Instead, the Examiner has simply separated elements of the above-recited claim features so as to find references that the Examiner believes disclose such elements. When determining differences between prior art and the claimed invention, "*the claimed invention as a whole must be considered*" (Emphasis added). Appellant asserts that the Examiner's analysis fails to consider the invention as a whole.

Further, although the Examiner asserts that "[i]t is a mere use of common sense by one skilled in the art to select and combine such known elements with no new function, i.e., a predictable result," as argued below, Appellant asserts that the cited references are not properly combinable to support a conclusion of obviousness with regard to the pending claims. As such, Appellant asserts the Examiner has not provided any combination of references that disclose storing "command line utility output in a system registry database at a location identified by the identifier."

In responding to Appellant's arguments in the Advisory Action, the Examiner stated:

> Appellant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references. Appellant's general argument about the application of the prior art of record to claim 1, 5, 10, 11, 15, 18, 21, and 23-25 was addressed in a previous action. Appellant's remarks regarding the combination of references will be addressed again below.

Advisory Action, page 2.

Appellant disagrees with the Examiner's argument, as Appellant is not making any general allegation that independent claims 1, 15, and 21 are allowable over the cited references. As argued above, Appellant asserts that the cited references do not disclose storing "command line utility output … at a location" and retrieving "the command line utility output … at the location identified by the identifier." Instead, the Examiner's rejection parses out those elements of the claims that the Examiner believes are found in the prior art. As noted above, Appellant objects to this characterization of the claims and argues that "the claimed invention as a whole must be considered." When the claims are afforded the correct consideration, Appellant asserts that the cited references do not disclose storing "command line utility output … at a location" and retrieving "the command line utility output … at the location identified by the identifier" as recited by claims 1, 15, and 21. Accordingly, Appellant respectfully requests the Board reverse the Examiner's rejection and allow claims 1, 15, and 21.

3. **The Examiner's rejection is improper because the combination of Buxton with Hill and Hlava is improper and cannot support a *prima facie* obviousness rejection.**

Appellant asserts that, for the reasons stated below, that the combination of Buxton, Hill, and Hlava is improper and cannot be used to reject the claims under 35 U.S.C. § 103(a). Appellant asserts that Buxton teaches away from combination with Hill and Hlava; that the combination of Buxton, Hill, and Hlava would change the principle of

operation of Hlava and renders Hlava unsuitable for its intended purpose; that Hlava clearly teaches away from combination with Buxton; and that the justification for the hypothetical combination of Buxton Hill and Hlava is an impermissible use of hindsight.

First, Appellant asserts that Buxton teaches away from a combination with Hill and Hlava. Hill is a reference directed to the "Windows NT Command Shell." Hill, page 1. The descriptions in Hill concern usage of the "command shell," a "command prompt," i.e., a command line, and various commands executed from the "command shell" by typing these commands into the "command prompt." *Id.* Similarly, Hlava is directed to "command files" that are described therein as "a file containing one or more command line operations." Hlava, col. 4, lines 10-20. Thus, both Hill and Hlava are directed to usage of the "command line" and various commands executed from the command line. In contrast, Buxton discloses "OLE libraries" that are defined as "system-level services which utilize the interfaces defined by the COM specification" that call a "WIN 32 API." Buxton, col. 8, lines 6-8. Appellant asserts that there is a clear difference between a service and the command executed from the command prompt as recited in Hill, and between a service and the command line operation as recited in Hlava. Further, as known to those of ordinary skill in the art and as stated in Buxton, API's are "application program interfaces" which are also quite different than a utility and a "command line utility." As they are described in Buxton, neither "application program interfaces" nor "system-level services" are "executable from a command line prompt," and thus cannot be considered a "command line utility." Appellant asserts one skilled in the art would not seek to combine Hill and Hlava, directed to command line usage, with Buxton, directed to usage of system-level services, e.g., OLE libraries. In view of this, Appellant asserts the "system level services" of Buxton are clearly different than a command line utility executed from the command prompt. System-level services, such as the "OLE libraries," are not "executable from a command line prompt," and cannot be considered a "command line utility." Appellant asserts one skilled in the art would not seek to combine Hill and Halva, directed to command line usage, with Buxton, directed to usage of system-level services, e.g., OLE libraries.

The Examiner stated that the combination is obvious because "both Hlava and Buxton involve storage of data in the registry" and "because of the advantages highlighted by Hlava as well as the particular implementation details of that registry provided by Buxton." Final Office Action mailed April 26, 2010, page 18. However, as noted by the Examiner, the advantages provided by Hlava are "allowing the command files to use the Windows registry." *Id.*, pages 17-18. However, this advantage *only* applies to such "command files" and the use thereof, and does not apply or provide any advantage to the system of Buxton. Hlava is directed to use of a "temporary command file" that is not operable with the techniques of Buxton. *See* Hlava, Abstract.

Second, Appellant asserts that not only does Hlava not disclose the features of independent claims 1, 15, and 21, but it teaches away from the claimed invention. As amended, independent claims 1, 15, and 21 recite storing command line utility output in a "system registry database." Appellant notes that Hlava maintains use of a "temporary command file." As previously stated, Appellant's claims are directed to providing command line utility output to applications without the need for temporary files. Specification, lines 16-17. Further, because the combination of Hill, Buxton and Hlava would have to remove the temporary file of Hlava in order to obviate claims 2-14, 16-20, and 23-25, Appellant asserts that such a modification would change the "principle of operation" of Hlava and render Hlava "unsuitable for its intended purpose." *See* M.P.E.P. 2143.01.

For example, in the rejection, the Examiner relied on the combination of the techniques of Hill, directed to a file, with the use of the "environment variables" disclosed in Hlava. *See* Final Office Action, pages 5-6 and 19-20. Appellant respectfully disagrees with the Examiner's interpretation. In particular, the Examiner is misinterpreting the environment variables of Hlava and the "variables described to temporary locations" of Hill. First, the portion of Hill cited by the Examiner does not describe or include any variables. Hill describes redirecting data to a file or to another command using the redirection and pipe commands. Hill does not discuss or mention

redirecting output to variables. Hill, pages 10-11. Second, the "environment variables" of Hlava are a specific type of variables used to store application configuration information. Hlava, col. 1, lines 27-34. There is no discussion in Hill of any of these environment variables or how the commands disclosed therein interact with environment variables. The environment variables of Hlava are stored in the Windows registry, a type of data store. *See id.i*, col. 2, lines 37-44. The techniques of Hill are directed to storing the output of command line utilities, such as "dir," in a text file. If the files of Hill were interchangeable with the "environment variables" of Hlava, there would be no need for the invention described in Hlava to use "environment variables". However, because such files are not suitable, Hlava is directed to accessing the registry to allow "full use of the data store as intended" as a "central repository for configuration type information." *Id.*, col. 2, lines 44-46. Substituting the use of a "file" would alter the intended purpose of Hlava and render it unsatisfactory for this purpose of allowing full use of the data store, *e.g.*, registry. Thus, to the extent that the Examiner's rational underpinning for the combination relies on the commonality between the "variables" of Hill and the "environment variables" described in Hlava, Appellant asserts there is no such commonality or basis for the combination, as Hill does not teach or suggest "variables," let alone the environment variables described in Hill. One of ordinary skill in the art would not seek to use the commands of Hill, such as redirection commands and pipe commands, with the system of Hlava that is clearly directed to the interaction between environment variables and the Windows Registry.

Third, in response to the Examiner's characterization of Buxton, Appellant notes that Hlava itself teaches away from combination with Buxton. The invention of Buxton clearly is directed to the "system-level service," such as "OLE libraries," that provide a different technique than that of Hlava. Buxton, col. 8, lines 6-8. Buxton is directed to objects used by applications or programs in the context of object-oriented programming techniques. *See Id.*, col. 2, lines 18-34. Appellant notes that Buxton states that the OLE libraries function through the use of WIN32APIs. *Id.*, col. 8, lines 8-9. In contrast,

Hlava clearly teaches away from accessing the registry through programs and Windows APIs. Hlava states:

> To provide access to Registry data, utility functions can be coded as programs rather than command files. These programs can then access the Registry data by using the Windows APIs. However, this method is somewhat inefficient. Programs are more difficult and time consuming to write and maintain than command files. Therefore, this method detracts from developer productivity relative to a method that uses command files to access the Registry data.
>
> Hlava, col. 1, lines 44-53. (Emphasis added.)

Thus, as noted above, Hlava discloses the use of "command files" as an alternative to registry access through programs using Windows APIs, such as in the techniques of Buxton. Even though the "command files" of Hlava are described as using APIs, Hlava provides this as an alternative to the use of programs and the APIs, such as would be used in the object-oriented programming techniques of Buxton. Accordingly, Appellant asserts that Hlava teaches away from combination with Buxton. Thus, Appellant asserts that one of ordinary skill in the art would not seek to combine the techniques of Hlava, directed to "command files" as an alternative to programs and Windows APIs, with the techniques of Buxton, which rely on OLE libraries and Windows APIs accessible by programs.

In responding to the above arguments in the Advisory Action, the Examiner stated:

> Examiner respectfully disagrees. As the applicant notes above, the Examiner has previously stated that Hlvana recognizes the advantage of storage in the Windows registry. Hlvana uses this advantage in the context of command files, which while different Buxton, is not inoperable. Buxton does not teach away from the combination of it's own registry elements with the command files & registry of Hlvana simply because

> Buxton uses an OLE library (which again, the examiner does not rely on for any element of the rejection) rather than a command file because "the prior art's mere disclosure of more than one alternative does not constitute a teaching away from any of these alternatives because such disclosure does not criticize, discredit, or otherwise discourage the solution claimed...." In re Fulton, 391 F.3d 1195, 1201, 73 USPQ2d 1141, 1146 (Fed. Cir. 2004).
>
> Further, with respect to applicant's arguments about Hlvana disparaging a combination with the registry API of Buxton, it is again emphasized that the examiner does not rely on the system level service or OLE libraries of Buxton, but instead only the system registry identification elements of Buxton. Therefore, regardless to Hlvana discussion of coding as programs rather than command files, Hlvana does not teach away from the combination as described in the rejection, because Hlvana is not disparaging the element relied on in the rejection, because the examiner does not rely on the system level services or OLE libraries of Buxton, but only the registry storage elements, which are not disparaged by Hlvana.

Advisory Action, pages 3-4.

In the Advisory Action, the Examiner disagreed with the Appellant's argument because the Examiner stated that he is not relying on the "system level services" or "OLE libraries" of Buxton. First, Appellant asserts the Examiner's reliance on *In Re Fulton* is misplaced, as *Fulton* refers to the disclosure of nonpreferred and alternative embodiments in a single prior art reference. *See* M.P.E.P. § 2123. Thus, when a reference discloses both a preferred and a nonpreferred embodiment, the disclosure of one alternative does not constitute teaching away from the other alternative. However, this legal guidance does not apply to the present rejection. Here, the Examiner is combining an embodiment from a first reference (Buxton) with an embodiment from a second reference (Hlava), even though the second reference explicitly discourages use of the embodiment from the first reference. Hlava does not teach that the techniques of Buxton are an alternate or nonpreferred embodiment. Instead, Hlava explicitly disclaims the use of the programs

and APIs taught by Buxton. *See* Hlava, col. 1, lines 44-53. Appellant asserts that the Examiner has not provided sufficient evidence or reasoning to show that one skilled in the art would ignore this clear disparagement of the technique of Buxton. When faced with such an explicit teaching away from Buxton, Appellant asserts that one skilled in the art would not seek to combine the "command files" technique of Hlava with the techniques of Buxton.

Finally, with regard to the Examiner's implication that the redirection of Hill may be used to redirect command line output to the registry, based on the techniques of Hlava, Appellant asserts that such a conclusion is an impermissible use of hindsight. *See* Final Office Action, pages 5-6. Such a conclusion is merely a restatement of Appellant's disclosed invention. That is, as previously argued, Appellant's claims are directed to storing the output of a command line utility in a "system registry database." The redirect command in Hill is inoperable for storing output in a "system registry database," but can only store data in a file. *See* Hill, pages 10-11. As noted above, Hlava provides "command files" for storing data in environment variables stored in a registry, but Hlava does not provide any evidence that such techniques are operable or suitable for use with storing command line utility output in a file. *See* Hlava, col. 1, lines 44-53. Thus, Appellant asserts that the reason for the combination is based on hindsight reconstruction gleaned from Appellant's invention that enables storing of the output of a "command line utility in a "system registry database."

In responding to Appellant's arguments, the Examiner stated:

> Examiner again respectfully disagrees. In response to applicant's argument that the examiner's conclusion of obviousness is based upon improper hindsight reasoning, it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See In re

> McLaughlin, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971). Appellant's arguments with regards to Buxton have been addressed above. Here, Hlvana's environmental variables provide separate storage in order to access elements that are subsequently stored in the registry (see. e.g. 340, FIG. 3 of Hlvana). Specifically, they environmental variables are not the registry itself, but are used to access the registry. (340, FIG. 3). Therefore, contrary to applicant's assertion, rather than making Hlvana unsatisfactory for its intended purpose, the combination with hill's "dir.txt" file, furthers Hlvana's purpose of using the environmental variables of Hlvana to provide ultimate access to the storage of the registry. (see. E.g. FIG. 2 of Hlvana).

Advisory Action, page 4.

Appellant respectfully asserts that the Examiner's response does not overcome Appellant's previous arguments with the regard to the impropriety of the combination. As noted above, Appellant asserts that the knowledge used for the combination was gleaned from the Appellant's disclosure, as Appellant's disclosure clearly distinguishes between the use of a temporary file and the techniques recited in the pending claims. *See* Application, pages 1-3. Further, with regard to the combination of the" environment variables" of Hlava with the "dir.txt" file of Hill, the Examiner has not provided any evidence or reasoning that links the use of "environment variables" that store data in a registry with a redirect command that redirects output to a file, as disclosed in Hill. Again, Appellant maintains that the only possible link between these techniques is gleaned from Appellant's specification which clearly describes a system that overcomes the inoperability of the hypothetical combination of Hlava and Hill.

For at least these reasons, Appellant asserts the hypothetical combination of Buxton, Hill, and Hlava is improper. Accordingly, Appellant respectfully requests the Board reverse the Examiner's rejection under 35 U.S.C. § 103.

4. **The Examiner's rejection of dependent claims 26-28 is improper because the rejection fails to establish a *prima facie* case of obviousness with regard to claims 26-28.**

Claim 26 recites "without creating a temporary file," claim 27 recites "without use of a temporary file," and claim 28 recites "without using a temporary file." Appellant asserts that the cited references, taken alone or in hypothetical combination, do not disclose these elements of claims 26-28.

In rejecting claims 26, 27, and 28, the Examiner stated:

> Here, note that nowhere does Hill suggest that the DIR.txt must be temporary. Further, with respect to the Hlvana reference, while Hlvana describes using temporary variables, Hlvana contemplates the use of both temporary and non-temporary variables as evidenced by, e.g. Claim 2 of Hlvana, where information is stored in an "environment variable" as opposed to the "temporary environment variables" of claim 4 in Hlvana.

Final Office Action mailed April 26, 2010, page 10.

However, the use the "dir.txt" file describes the prior art technique distinguished in the background of the present application. Indeed, the present application specifically mentions "dir" as one of the "illustrative command line utilities." Application, page 1, lines 11-12. The "dir.txt" file referred to in Hill is necessary for the storage and use of the results of the "dir" command. Thus, to the extent that the results of the "dir" command are to be stored and used by an executing application, the "dir.txt" is a temporary file necessary for such use. The Examiner also noted that Hlava "contemplates the use of temporary and non-temporary variables." Final Office Action mailed April 26, 2010, page 10. As argued elsewhere in this Brief, Appellant asserts that Hlava is deficient for other reasons and is not properly combinable with Buxton and Hill. Thus, Appellant respectfully requests that the Board reverse the rejection of claims 26-28 under 35 U.S.C. § 103(a).

In the Advisory Action, the Examiner stated:

> Examiner respectfully disagrees. In response to applicants argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e. the limiting meaning of "temporary" other than temporal) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See In re Van Geuns, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Advisory Action, page 3.

Appellant maintains the term "temporary file" should be given its "broadest reasonable interpretation consistent with the specification" and "the interpretation that those skilled in the art would reach." *See* M.P.E.P. § 2111. Accordingly, Appellant notes that the "temporary file" is temporary for purposes of storing the output of a command line utility and using that output by an executing application. *See* Application, page 1, lines 16-27. Thus, the term "temporary" refers to the use of the file with regard to the use in the described prior art technique, not necessarily in the temporal aspect of the file. Such an interpretation is consistent with the specification and the use of such a term by those of ordinary skill in the art in the context of command line utilities. In view of this correct interpretation, Appellant asserts that the cited references, taken alone or in hypothetical combination, fail to disclose the claim elements of "without creating a temporary file" as recited in claim 26, "without use of a temporary file" as recited in claim 28, and "without using a temporary file" as recited in claim 28. Accordingly, Appellant respectfully requests the Board reverse the Examiner's rejection of claims 26-28 and allow the claims.

B. **Ground of Rejection No. 2:**

The Examiner rejected claims 2-4, 7, 12-14, 19 and 20 under 35 U.S.C. § 103(a) as being unpatentable over AAPA in view of Hill in view of Hlava as applied above and further in view of Buxton. Appellant respectfully traverses this rejection.

1. **The Examiner's rejection of dependent claims 26-28 is improper because the rejection fails to establish a *prima facie* case of obviousness with regard to claims 26-28.**

Claims 2-4, 7, and 12-14 are dependent on claim 1 and claims 19 and 20 are dependent on claim 15. As discussed above with regard to the first ground of rejection under 35 U.S.C. § 103(a), the cited references, taken alone or in hypothetical combination, do not disclose all claimed features of claims 1 and 15. Accordingly, the cited combination does not disclose or suggest all of the elements of the claimed invention, and thus, cannot possibly render the claimed subject matter obvious. Thus, Appellant respectfully requests the Board reverse the Examiner's rejection of claims 2-4, 7, 12-14, 19, and 20 under 35 U.S.C. § 103(a) and allow the claims.

Conclusion

Appellant respectfully submits that all pending claims are in condition for allowance. However, if the Examiner or Board wishes to resolve any other issues by way of a telephone conference, the Examiner or Board is kindly invited to contact the undersigned attorney at the telephone number indicated below.

Respectfully submitted,

Date: October 26, 2010

/Robert A. Manware/

Robert A. Manware
Reg. No. 48,758
FLETCHER YODER
P.O. Box 692289
Houston, TX 77269-2289
(281) 970-4545

8. **APPENDIX OF CLAIMS ON APPEAL**

1. A method comprising:

invoking, by an application, a call of a command line utility, the application providing an identifier in the call of the command line utility, wherein the command line utility is a utility executable from a command line prompt;

receiving output from the command line utility;

storing the command line utility output in a system registry database at a location identified by the identifier; and

retrieving, by the application, the command line utility output from the system registry at the location identified by the identifier.

2. The method of claim 1, wherein providing the identifier comprises providing an identifier that identifies one or more entries in the system registry database.

3. The method of claim 2, wherein providing the identifier comprises providing a root key identifier.

4. The method of claim 3, wherein providing the root key identifier comprises providing a sub-key identifier.

5. The method of claim 1, wherein the system registry database comprises an operating system registry database.

7. The method of claim 1, wherein providing the identifier comprises providing an identifier indicating the system registry database.

10. The method of claim l, wherein the act of receiving output from a command line utility comprises receiving output directly from the command line output utility.

11. The method of claim 1, wherein the act of receiving output from a command line utility comprises receiving output from the command line output utility through a subsequent command line output routine.

12. The method of claim 1, wherein the act of storing comprises associating each line of command line utility output with a line identifier in the system registry database.

13. The method of claim 12, further comprising setting each line identifier to a value corresponding to a position of that line in the command utility output.

14. The method of claim 12, further comprising setting a default value of the provided identifier to equal the total number of command utility output lines stored in the system registry database.

15. A program storage device, readable by a computer, comprising instructions stored on the program storage device for causing the computer to:

cause an application to invoke a call of a command line utility, the application providing an identifier in the call of the command utility, wherein the command line utility is a utility executable from a command line prompt;

receive output from the command line utility;

store the command line utility output in a system registry database at a location identified by the identifier; and

cause the application to retrieve the command line utility output from the system registry database or shared system memory at the location identified by the identifier.

18. The program storage device of claim 15 wherein the instructions to receive output comprise instructions to receive one or more lines of output from the command line utility, and the instructions to store further comprise instructions to store each of said one or more lines of output in the system registry database.

19. The program storage device of claim 18 wherein the instructions to store further comprise instructions to associate a unique identifier with each of the one or more lines of output stored in the system registry database.

20. The program storage device of claim 18 wherein the instructions to store further comprise instructions to set a value associated with the received identifier in the system registry database or shared system memory equal to the number of lines of output stored in the system registry database.

21. A computer system, comprising:

a processor;

a command line utility, wherein the command line utility is a utility executable from a command line prompt;

an application executable on the processor, the application to call the command line utility, the application to provide an identifier in the call;

a system registry database having a location identified by the identifier, the location identified by the identifier to store an output of the command line utility,

the application to retrieve the command line utility output from the location identified by the identifier.

23. The method of claim 1 wherein the command line utility comprises a first command line utility, and wherein invoking the call by the application comprises invoking a call to pipe output of a second command line utility to the first command line utility,

wherein storing the command line utility output comprises storing the command line utility output of the first command line utility.

24. The program storage device of claim 15, wherein the command line utility comprises a first command line utility, and. wherein invoking the call by the application

comprises invoicing a call to pipe output of a second command line utility to the first command line utility,

wherein storing the command line utility output comprises storing the command line utility output of the first command line utility.

25. The computer system of claim 21, wherein the command line utility comprises a first command line utility, the system further comprising a second command line utility, the application to invoke a call that causes output of the second command line utility to be piped to the first command line utility,

the location identified by the identifier to store output of the first command line utility.

26. The method of claim 1, wherein receiving output from the command line utility comprises receiving output without creating a temporary file.

27. The program storage device of claim 15, wherein the instructions stored on the program storage device further cause the computer to receive output from the command line utility without use of a temporary file.

28. The computer system of claim 21, wherein the location identified by the identifier stores the output of the command line utility without using a temporary file.

9. **EVIDENCE APPENDIX**

None.

10. **RELATED PROCEEDINGS APPENDIX**

None.